UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/670,068	09/23/2003	Akihiko Tozawa	JP920020151JP1	5679

36380 7590 07/27/2007

RICHARD M. GOLDMAN
371 ELAN VILLAGE LANE
SUITE 208, CA 95134

EXAMINER	
PHAM, MICHAEL	

ART UNIT	PAPER NUMBER
2167	

MAIL DATE	DELIVERY MODE
07/27/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/670,068	TOZAWA ET AL.	
	Examiner	Art Unit	
	Michael D. Pham	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 3.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1,3,4,6,10,11,14,15,18-20 and 22 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1,3,4,6,10,11,14,15,18-20 and 22 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☒ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Detailed Action

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 5/6/2007 has been entered.

Status of claims

2. Claims *1,3,4,6,10,11,14,15,18-20 and 22* are pending.

3. Claims *1,3,4,6,10,11,14,15,18-20 and 22* have been examined.

Priority

4. Should applicant desire to obtain the benefit of foreign priority under 35 U.S.C. 119(a)-(d) prior to declaration of an interference, a certified English translation of the foreign application must be submitted in reply to this action. 37 CFR 41.154(b) and 41.202(e).

Failure to provide a certified translation may result in no benefit being accorded for the non-English application.

Specification

5. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the

following is required: Claims 1, 4, 10, 14, 18, and 20 are objected because there is no clear concise meaning for the term axis.

37 CFR 1.75(d)(1):

(1) The claim or claims must conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the claims must find clear support or antecedent basis in the description so that the **meaning of the terms** in the claims may be ascertainable by reference to the description. (See § **1.58**(a)).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1, 3-4, 6, 10, 11, 14-15, 18, 19-20, and 22 rejected under 35 U.S.C. 103(a) as being unpatentable over "Efficient Filtering of XML Documents for Selective Dissemination of Information", International Conference on Very Large Data Bases, 2000, by Altinel, Mehmet and Franklin, Michael J., (Hereinafter, Altinel) as further in view of U.S. Patent 6,052,686 by Fernandez et. al. (Hereafter, Fernandez).

Claim 1:

Atinel discloses the following claimed limitations:

"A compiling device for generating a query automation by storing an input query expression, performing parsing, identifying different types of nodes in said element identifiers" [See page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries..." and see pages 56-57 "Each XPath query is decomposed into a set of path nodes by the XPath parser. These path nodes represent the element nodes in the query and serve as the states of the FSM for the query." In other words, the documents are converted into the query automaton, identified, and stored.]

"A query automaton storage device storing the query generated by said compiling device;" [See page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries..."]

"A query automaton evaluator for reading out said query automaton from said storage device storing said automaton" [See page 54, column 1 "These profiles are 'standing queries,' which are (conceptually) applied to all incoming documents" In other words "applied to all incoming documents" is equivalent to "reading out."]

", while reading in said document and performing a stream search by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton" [See page 54, column 1, The other key inputs to an SDI system are the

documents to be filtered."; also see page 57, column 2 "We use an XML parser that is based on the SAX interface, which is a standard for event-based XML parsing. And see page 56, column 2, The Query Index is built over the states of the XPath queries. The SAX interface from the reference uses the stream search as described in the specification on page 3, line 1.]

"and outputting the searched node and a search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." ([See page 57, column 2, 4th paragraph, All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the FSM of a query is represented by promoting a path node from the Wait List to the Candidate List. The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph. If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.]

Altinel does not explicitly disclose, ",wherein the compiling device generates and registers a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query

equal in terms of search, wherein the compiling device generates a query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state;"

and

", said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node,".

On the other hand, Fernandez discloses col. 3 lines 19-22, a schema captures characteristics in a structural form and the schema maybe used to improve database processing efficiency (e.g. generates and registers). Further disclosing col. 3 line 23, schema 400 contains states S1-S12 (e.g. state transition by replacing axis including axis in opposite direction). That col. 3 lines 44-45, a transition from state S1 to S2 or S3. Col. 4 The automaton models the path expression and is equivalent to it. Fernandez further discloses col. 4 lines 53-54, the automaton models the path expression (e.g. and logical expression) and is equivalent to it (e.g. while keeping an input query equal in terms of search). That arrows labeled 'other' are simply the negation (e.g. including a negation) of all the edges; the other arrows emanates and terminates from/to S1 and S2 are equivalent to arrows labeled not(Org) AND not(Project) (e.g. including a conjunction). Further disclosing, figure 4, containing backwards nodes, condition for transition, and search states, and further contains left nodes as well as lower nodes for the left nodes to determine the final state, (e.g. determining said query automaton with a search result of said left and said lower node).using conditions to transition to next state (e.g. element identifiers).

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary state machines to the system of Altinel's for the benefit of pruning or delivering prior materials through construction of a composite automaton, in doing so, it provides the benefit of reducing the search time.

Claim 3:

Fernandez discloses "wherein said compiling device generates a query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon "[figure 5, provides DB (e.g. initial state), A1 (e.g. search state), P (e.g. final state)].

Claim 4:

Atinel discloses the following claimed limitations:

"generating a query automaton by storing a query expression input by a compiling device performing parsing, and identifying different types of nodes in said element identifiers;" [See page 55, column 2, In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries... and see page 56-57 Each XPath query is decomposed into a set of path nodes by the XPath parser. These path nodes represent the element nodes in the query and serve as the states of the FSM for the query."]

"storing the query automaton generated by said compiling device in a query automaton storage device; and" [See page 55, column 2, In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries... In other words, the documents are converted into the query automaton, identified, and stored.]

"reading out said query automaton from said query automaton storage device and storing said query automaton" [See page 54, column 1, These profiles are 'standing queries,' which are (conceptually) applied to all incoming documents]

", while reading in said document and performing a stream search with a query automaton evaluator by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton, and" [See page 54, column 1, The other key inputs to an SDI system are the documents to be filtered. Also see page 57, column 2, We use an XML parser that is based on the SAX interface, which is a standard for event-based XML parsing...and see page 56, column 2 The Query Index is built over the states of the XPath queries. The SAX interface from the reference uses the stream search as described in the specification on page 3, line 1.]

"storing the output of the query automaton evaluator in a search result storage means for, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." [See page 57, column 2, 4th paragraph "All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the

FSM of a query is represented by promoting a path node from the Wait List to the Candidate List." The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph "If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.]

However, Atinel does not explicitly disclose further "by the steps of generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, the query automaton including a plurality of states of the backwards nodes, a condition for transition, and at least a search state, said query automaton evaluator determining a state transition, of a node under determination of storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node"

On the other hand, Fernandez discloses col. 3 lines 19-22, a schema captures characteristics in a structural form and the schema maybe used to improve database processing efficiency (e.g. generates and registers). Further disclosing col. 3 line 23, schema 400 contains states S1-S12 (e.g. state transition by replacing axis including axis in opposite direction). That

col. 3 lines 44-45, a transition from state S1 to S2 or S3. Col. 4 The automaton models the path expression and is equivalent to it. Fernandez further discloses col. 4 lines 53-54, the automaton models the path expression (e.g. and logical expression) and is equivalent (e.g. while keeping an input query expression equal in terms of search) to it. That arrows labeled 'other' are simply the negation (e.g. including a negation) of all the edges; the other arrows emanates and terminates from/to S1 and S2 are equivalent to arrows labeled not(Org) AND not(Project) (e.g. including a conjunction). Further disclosing figure 4 contains backwards nodes, condition for transition, and search states, and further contains left nodes as well as lower nodes for the left nodes to determine the final state, (e.g. determining said query automaton with a search result of said left and said lower node).using conditions to transition to next state (e.g. element identifiers).

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary state machines to the system of Altinel's for the benefit of pruning or delivering prior materials through construction of a composite automaton, in doing so, it provides the benefit of reducing the search time.

Claim 6:

Fernandez discloses "wherein said step of generating a query automaton comprises a step of generating a query automaton with a state transition corresponding to an initial state, final state,

and a search state registered thereon"[for example figure 5, provides DB (e.g. initial state), A1 (e.g. search state), P (e.g. final state)].

Claim 10:

Atinel discloses the following claimed limitations:

"functioning as a compiling device for generating a query automaton by storing an input query expression, performing parsing, and identifying different types of nodes in said element identifiers," [See page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries..." and see page 56-57 "Each XPath query is decomposed into a set of path nodes by the XPath parser. These path nodes represent the element nodes in the query and serve as the states of the FSM for the query." In other words, the documents are converted into the query automaton, identified, and stored.]

"storing a query automaton generated by said compiling device in a query automaton storage device; "[See page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries..."]

"functioning as a query automaton evaluator for reading out said query automaton from said storage device and storing said query automaton," [See page 54, column 1 "These profiles are 'standing queries,' which are (conceptually) applied to all incoming documents"]

"while reading in said document and storing said query automaton, while reading in said document and performing a stream search by using states of a plurality of different types of nodes in said element identifiers included in said document and said query automaton and" [See page 54, column 1 "The other key inputs to an SDI system are the documents to be filtered." also see page 57, column 2 "We use an XML parser that is based on the SAX interface, which is a standard for event-based XML parsing..." and see page 56, column 2 "The Query Index is built over the states of the XPath queries." The SAX interface from the reference uses the stream search as described in the specification on page 3, line 1.]

"storing the output of the query automaton evaluator in a search result storage means, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." [See page 57, column 2, 4th paragraph "All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the FSM of a query is represented by promoting a path node from the Wait List to the Candidate List." The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph "If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.]

However, Atinel does not explicitly disclose further "by the steps of generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, the query automaton including a plurality of states of the backward nodes, a condition for transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node;"

On the other hand, Fernandez discloses col. 3 lines 19-22, a schema captures characteristics in a structural form and the schema maybe used to improve database processing efficiency (e.g. generates and registers). Further disclosing col. 3 line 23, schema 400 contains states S1-S12 (e.g. state transition by replacing axis including axis in opposite direction). That col. 3 lines 44-45, a transition from state S1 to S2 or S3. Col. 4 The automaton models the path expression and is equivalent to it. Fernandez further discloses col. 4 lines 53-54, the automaton models the path expression (e.g. and logical expression) and is equivalent (e.g. while keeping an input query expression equal in terms of search) to it. That arrows labeled 'other' are simply the negation (e.g. including a negation) of all the edges; the other arrows emanates and terminates from/to S1 and S2 are equivalent to arrows labeled not(Org) AND not(Project) (e.g. including a conjunction). Further disclosing figure 4 contains backwards nodes, condition for transition, and search states, and further contains left nodes as well as lower nodes for the left nodes to determine the final state, (e.g. determining said query automaton with a search result of said left and said lower node).using conditions to transition to next state (e.g. element identifiers).

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary state machines to the system of Altinel's for the benefit of pruning or delivering prior materials through construction of a composite automaton, in doing so, it provides the benefit of reducing the search time.

Claim 11:

Fernandez discloses "wherein said performance of stream search determines a state transition of a node under determination at the moment by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node, and wherein said query automaton is generated as a query automaton with a state transition corresponding to an initial state, a final state, and a search state registered thereon "[for example figure 5, provides DB (e.g. initial state), A1 (e.g. search state), P (e.g. final state). That the nodes are stored and only transition when a condition is met, thereby suggesting wherein in said performance determines a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier and evaluating the query automaton with a search result of the left and lower node.].

Claim 14:

Atinel discloses the following claimed limitations:

"storing the output of the query automaton evaluator in a search result storage means for, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." (See page 57, column 2, 4th paragraph "All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the FSM of a query is represented by promoting a path node from the Wait List to the Candidate List." The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph "If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.)

Altinel does not explicitly disclose

"generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, and storing a plurality of states

of a backward node in correspondence with said backward node into a query automaton storage device;"

"generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of generating and registering a state transition by the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state, said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node and"

On the other hand, Fernandez discloses "generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, and storing a plurality of states of a backward node in correspondence with said backward node into a query automaton storage device;" [col. 3 lines 19-22, a schema captures characteristics in a structural form and the schema maybe used to improve database processing efficiency (e.g. generates and registers). col. 3 line 23, schema 400 contains states S1-S12 (e.g. state transition by replacing axis including axis in opposite direction). The automaton models the path expression and is equivalent to it (e.g. while keeping an input query expression equal in terms of search). Fernandez further discloses col. 4 lines 53-54, the automaton models the path

expression (e.g. logical expression). Among others, Figure 4, schema contains states (e.g. discloses storage of states of a backward node in correspondence with said backward node into a query automaton storage device).]

"generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of generating and registering a state transition by the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state, said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node and"[Col. 4 lines 33-36, path expression corresponds to an automaton 500 as shown in figure 5. The automaton 500 has an initial state labled DB The DB state indicates the starting point when traversing the automaton based on the path expression. Col. 4 lines 38-41, if a dept edge is encountered in the database as modeled by the edge-labled tree, then the automaton transitions from the db state to an A1 state. Col. 4 lines 50-51 the p state is a terminal state of the automaton 500. (e.g. backward node states (nodes), a condition for transition (expression), search state (traversal of nodes), and reach state (a terminal state).). col. 3 lines 19-20, schema captures characteristics of edge labeled tree in a structural form (i.e. parse). Col. 4 lines 52-53, automaton responds (evaluates) to a string sequence that is identical to the path expression. Figure 5, further contains left nodes as well as lower nodes for the left nodes to

determine the final state, (e.g. determining said query automaton with a search result of said left and said lower node).using conditions to transition to next state (e.g. element identifiers)]

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary states to the system of Altinel's for the benefit of pruning, in doing so, it provides the benefit of reducing the search time.

Claim 15:

Fernandez discloses "identifying said backward node as a left node or a lower node according to a type of said element identifier, and wherein said plurality of states are states of said left node and said lower node." [DB, A1 (e.g. backward node as a left node or a lower node according to a type of said element identifier (condition)), and P states.]

Claim 18:

Altinel discloses the following claimed limitations:

"storing the output of the query automaton evaluator in a search result storage means, and thereafter outputting the stored output of the query automaton evaluator and the output of the searched node."

Altinel does not explicitly disclose,

"generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, and storing a plurality of states of a backward node in correspondence with said backward node into a query automaton storage device;"

"generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of generating and registering a state transition by the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state, said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node and"

On the other hand, Fernandez discloses "generating and registering a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, and storing a plurality of states of a backward node in correspondence with said backward node into a query automaton storage device;" [col. 3 lines 19-22, a schema captures

characteristics in a structural form and the schema maybe used to improve database processing efficiency (e.g. generates and registers). col. 3 line 23, schema 400 contains states S1-S12 (e.g. state transition by replacing axis including axis in opposite direction). The automaton models the path expression and is equivalent to it (e.g. while keeping an input query expression equal in terms of search). Fernandez further discloses col. 4 lines 53-54, the automaton models the path expression (e.g. logical expression). That arrows labeled 'other' are simply the negation (e.g. including a negation) of all the edges; the other arrows emanates and terminates from/to S1 and S2 are equivalent to arrows labeled not(Org) AND not(Project) (e.g. including a conjunction). Among others, Figure 4, schema contains states (e.g. discloses storage of states of a backward node in correspondence with said backward node into a query automaton storage device).]

"generating a query automaton by registering a plurality of states of said backward node, a condition for transition, at least a search state, and a reached state in correspondence with each other in said storage device, performing parsing, and identifying different types of nodes in said element identifiers, by the steps of generating and registering a state transition by the query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state, said query automaton evaluator determining a state transition of a node under determination by storing a left node and a lower node in correspondence with an identified element identifier, and evaluating said query automaton with a search result of said left node and said lower node and"[Col. 4 lines 33-36, path expression corresponds to an automaton 500 as shown in figure 5. The automaton 500 has an initial state labled DB The DB state indicates the starting point when traversing the automaton based on the path expression. Col. 4 lines 38-41, if a dept edge is encountered in the database as modeled by the edge-labled tree, then the

automaton transitions from the db state to an A1 state. Col. 4 lines 50-51 the p state is a terminal state of the automaton 500. (e.g. backward node states (nodes), a condition for transition (expression), search state (traversal of nodes), and reach state (a terminal state).). col. 3 lines 19-20, schema captures characteristics of edge labeled tree in a structural form (i.e. parse). Col. 4 lines 52-53, automaton responds (evaluates) to a string sequence that is identical to the path expression. Figure 5, further contains left nodes as well as lower nodes for the left nodes to determine the final state, (e.g. determining said query automaton with a search result of said left and said lower node).using conditions to transition to next state (e.g. element identifiers)]

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary states to the system of Altinel's for the benefit of pruning, in doing so, it provides the benefit of reducing the search time.

Claim 19:

Fernandez discloses "identify said backward node as a left node or a lower node according to a type of said element identifier, and wherein said plurality of states are states of said left node and

said lower node." [DB, A1 (e.g. backward node as a left node or a lower node according to a type of said element identifier (condition)), and P states.]

Claim 20:

Altinel discloses the following claimed limitations:

"a compiling device for generating a two-state input query automaton for enabling a state transition by storing an input query expression, performing parsing," (See page 55, column 2, In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries. And see page 56-57, Each XPath query is decomposed into a set of path nodes by the XPath parser. These path nodes represent the element nodes in the query and serve as the states of the FSM for the query." In other words, the documents are converted into the query automaton and stored.)

"an automaton query storage device for storing said two-state input automaton;" (See page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries...")

"an automaton-evaluating device for enabling a state transition by reading out two-state input automaton from said storage device and storing said automaton" (See page 54, column 1,

These profiles are 'standing queries,' which are (conceptually) applied to all incoming documents)

", while reading in said document and identifying said two states node, and performing a stream search by using states of a plurality of different types of nodes in the element identifiers included in the document and the query automaton and" (See page 54, column 1, The other key inputs to an SDI system are the documents to be filtered.; also see page 57, column 2, We use an XML parser that is based on the SAX interface, which is a standard for event-based XML parsing. And see page 56, column 2, The Query Index is built over the states of the XPath queries. The SAX interface from the reference uses the stream search as described in the specification on page 3, line 1.)

"a search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." (See page 57, column 2, 4th paragraph "All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the FSM of a query is represented by promoting a path node from the Wait List to the Candidate List." The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph "If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the

promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.)

However, Altinel does not explicitly disclose "and reading at least two states assigned to different types of nodes in said element identifiers, wherein the compiling device generates and registers a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search, wherein the compiling device generates a query automaton including a plurality of states of the backward nodes a condition for transition, and at least a search state, and wherein said two states are states of a left node and a lower node of a tree structure generated in correspondence with an identified element identifier, and wherein said two-state input automaton uses three states of said automaton-evaluating device;"

On the other hand, Fernandez discloses "and reading at least two states assigned to different types of nodes in said element identifiers," [figure 5, conditions (identifier of next node state) cause transition between states, hence reading at least two states assigned to different types of nodes] "wherein the compiling device generates and registers a state transition by replacing an axis including an axis in the opposite direction and a logical expression including a conjunction or a negative expression while keeping an input query expression equal in terms of search" [col. 3 lines 19-21, a schema captures characteristics (e.g. generates and registers a state transition by replacing an axis including an axis in the opposite direction). Fernandez further discloses col. 4 lines 53-54, the automaton models the path expression (e.g. and logical expression) and is

equivalent (e.g. while keeping an input query expression equal in terms of search) to it. Col. 5 lines 1-4, negation (negation expression) of all the edges already discussed above. Thus the other arrows that emanates and terminates from/to S1 and S2 are equivalent to arrow labeled not(ORG) AND not(Project) (e.g. conjunction expression)], "wherein the compiling device generates a query automaton including a plurality of states of the backward nodes, a condition for transition, and at least a search state," [figure 5 among others, a plurality of states of the backwards nodes such as DB and A1. A condition for transition (satisfied condition transition to next state such as DB to A1). A search state such as A1 searches for the next state until it finds condition for P which is the terminating state.] "and wherein said two states are states of a left node and a lower node of a tree structure generated in correspondence with an identified element identifier, and wherein said two-state input automaton uses three states of said automaton-evaluating device;" [figure 5 among others, discloses two states are states of a left node and a lower node of a tree structure such as state DB and A1 which are generated in correspondence to conditions. In other words, DB and A1 are left nodes and lower nodes of a tree structure and depend on conditions to identify states. The two state input automaton uses three states such as DB, A1, and P.]

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary states to

the system of Altinel's for the benefit of pruning, in doing so, it provides the benefit of reducing the search time.

Claim 22:

Altinel discloses the following claimed limitations:

"Means for reading out a query automaton from a query automaton storage device and storing the query automaton," [See page 54, column 1 "These profiles are 'standing queries,' which are (conceptually) applied to all incoming documents" In other words "applied to all incoming documents" is equivalent to "reading out." and see page 55, column 2 "In contrast, in SDI systems, large numbers of queries are stored, and the documents are individually mated to the queries..."]

"while reading in the document and performing a stream search by using states of a plurality of different types of nodes in the element identifiers including in the document and the query automaton" [See page 54, column 1 "The other key inputs to an SDI system are the documents to be filtered." also see page 57, column 2 "We use an XML parser that is based on the SAX interface, which is a standard for event-based XML parsing..." and see page 56, column 2 "The Query Index is built over the states of the XPath queries." The SAX interface from the reference uses the stream search as described in the specification on page 3, line 1.]

"A search result storage means for storing the output of the query automaton evaluator, and for thereafter outputting the stored output of the query automaton evaluator and the output of the searched node." [See page 57, column 2, 4th paragraph "All of the path nodes representing future states are stored in the Wait Lists of their respective element names. A state transition in the FSM of a query is represented by promoting a path node from the Wait List to the Candidate List." The lists here are interpreted to be the storage means for storing the output of the query automaton. And see page 58, first column, 3rd full paragraph "If this is the final path node of the query (i.e., its final state) then the document is deemed to match the query. Otherwise, if it is not the final node, then the query is moved into its next state. This is done by copying the next node for the query from its Wait List to its corresponding Candidate List (note that a copy of the promoted node remains in the Wait List)." This determination represents the output of the query automaton evaluator.]

However Altinel does not explicitly disclose

"Means for identifying two different types of nodes included in said document wherein states of a left node and a lower node in a tree structure are generated in correspondence with an identified element identifier;"

"Means for assigning a state transition among three states including a search state by using said identified input and a plurality of inputs registered in said query automaton node and"

On the other hand, Fernandez discloses

"Means for identifying two different types of nodes included in said document wherein states of a left node and a lower node in a tree structure are generated in correspondence with an identified element identifier;" [figure 5, DB and A1 for example disclose two types of nodes generated in correspondence with an identified element identifier]

"Means for assigning a state transition among three states including a search state by using said identified input and a plurality of inputs registered in said query automaton node and" [figure 5, DB, A1, and P for example disclose assigning state transition among three states. That the states include a searching state by using identified input and a plurality of inputs registered such as the conditions in node A1]

Fernandez and Altinel are related to reducing of search and further are systems relating to state machines. And are therefore within the same field of endeavor. It would have been obvious to a person of an ordinary skill in the art to at the time the invention was made to apply the teachings of Fernandez of providing structures that capture and render the necessary states to the system of Altinel's for the benefit of pruning, in doing so, it provides the benefit of reducing the search time.

Response to Arguments

8. Applicant's arguments with respect to claims 1, 3-4, 6, 10, 11, 14-15, 18, 19-20, and 22 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

9. The prior art made of record listed on PTO-892 and not relied, if any, upon is considered pertinent to applicant's disclosure.

Contact Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael D. Pham whose telephone number is (571)272-3924. The examiner can normally be reached on Monday - Friday 9am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Michael Pham
Art Unit 2167
Examiner M.P.

Cam Y. Truong
Art Unit 2162
Primary Examiner
CT

John Cottingham
Art Unit 2167
Supervisor

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100